Amendments to the Claims:

The following listing of the claims will replace all prior versions and listings of claims in the application.

1 – 5 (Cancelled).

6. (New) A computer implemented method of detecting privilege escalation vulnerabilities in a pre-existing source code listing, said source code listing having a listed sequence of expressions, each expression including a set of operands and operators to transform values of the operands, said source code listing further having routine calls, said routine calls including arguments with which to invoke a routine, said source code listing being stored in computer readable medium having computer executable instructions, wherein a privilege escalation vulnerability is an uncontrolled escalation of system privileges that allows unauthorized access to system resources, the computer implemented method comprising:

executing computer instructions to provide a list specifying routines that potentially cause privilege escalation vulnerabilities;

executing computer instructions to provide pre-specified ranges of values for arguments of routines in the list that cause privilege escalation vulnerabilities;

executing computer instructions to analyze the source code listing to identify calls to routines specified in the list;

executing computer instructions to analyze the source code listing to semantically analyze arguments of the identified routine calls to determine routine calls that possess privilege escalation vulnerabilities using the pre-specified ranges of values; and

executing computer instructions to generate a report that identifies the vulnerabilities.

7. (New) The method of claim 6 wherein executing computer instructions to semantically analyze the arguments of the identified routine calls comprises analyzing the source code listing

to create computer models of the arguments, each model specifying a range of values that each corresponding argument can take when the source code listing is executed, said argument models being stored in computer memory.

8. (New) The method of claim 7, wherein analyzing the source code listing to create computer models of the arguments comprises:

analyzing the source code listing to create computer models of said operands, each of said operand models specifying a range of values of each corresponding operand as a result of operand transformations expressed in the source code listing, said models being stored in computer memory; and

using the operand models to create the argument models.

9. (New) A computer implemented utility for detecting vulnerabilities in a pre-existing source code listing, said source code listing having a listed sequence of expressions, each expression including a set of operands and operators to transform values of the operands, said source code listing further having routine calls, said routine calls including arguments with which to invoke a routine, wherein a privilege escalation vulnerability is an uncontrolled escalation of system privileges that allows unauthorized access to system resources, said utility comprising a computer-readable medium encoded with:

computer-executable instructions to provide a list specifying routines that potentially cause privilege escalation vulnerabilities;

computer-executable instructions to provide pre-specified ranges of values for arguments of routines in the list that cause privilege escalation vulnerabilities;

computer-executable instructions to analyze the source code listing to identify calls to routines in the list;

computer-executable instructions to analyze the source code listing to semantically analyze arguments of the identified routine calls to determine routine calls that

possess privilege escalation vulnerabilities using the pre-specified ranges of values; and

computer-executable instructions to generate a report that identifies the vulnerabilities.

10. (New) The utility of claim 9 wherein the computer-executable instructions to semantically analyze the arguments of the identified routine calls comprises computer-executable instructions for analyzing the source code listing to create computer models of the arguments, each model specifying a range of values that each corresponding argument can take when the source code listing is executed, said argument models being stored in computer memory.

11. (New) The utility of claim 10, wherein computer-executable instructions for analyzing the source code listing to create computer models of the arguments comprises:

computer-executable instructions for analyzing the source code listing to create computer models of said operands, each of said operand models specifying a range of values of each corresponding operand as a result of operand transformations expressed in the source code listing, said models being stored in computer memory; and

computer-executable instructions for creating the argument models using the operand models.